



Αντώνης Χρυσόπουλος
Κατερίνα Κυπριώτη
Στέλιος Μόσχογλου
Θεοδόσης Σουργκούνης

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ PHP

PHP 3: The Revenge of PHP – Η Εκδίκηση

Εφημερίδες! Έκτακτο Παράρτημα!

1. Booleans – Null (κενό)
2. Πίνακες
3. Εντολή foreach
4. Διαχωρισμός σε αρχεία
5. Εμβέλεια Μεταβλητών
6. Συναρτήσεις Αρχείων
7. Διαχείριση Αρχείων
8. Headers (Κεφαλίδες)
9. Cookies (Μπισκότα) - Sessions (Συνοδοί)

Πρώτη Είδηση!

1. Booleans – Null (κενό)

Booleans

- Οι μεταβλητές τύπου `boolean` παίρνουν δύο τιμές:
 1. *true (1)*
 2. *false (0)*
- Στην εκτύπωση των τιμών αυτών από την PHP αυτό που βλέπουμε στην οθόνη είναι τα αριθμητικά τους ισοδύναμα
...ή μήπως όχι???

Παράδειγμα

```
<?php  
$var1 = true;  
$var2 = false;  
echo "var1: " . $var1;  
echo "var2: " . $var2;  
?>
```



var1: 1

var2: (????????)

Τι μαγικό ήταν αυτό???

- Στην περίπτωση που έχουμε *false*, δεν εμφανίζεται τίποτα στην οθόνη, ούτε *false* ούτε *0*.
- Οπότε δεν τρομάζουμε όταν συμβαίνει αυτό, είναι φυσιολογικό.



NULL (Κενό)

- Η λέξη *NULL* υποδηλώνει μια μεταβλητή που δεν περιέχει μέσα της καμία τιμή ούτε είναι κάποιου συγκεκριμένου τύπου, είναι απλά ορισμένη ως όνομα.
- Όταν εκτυπώνουμε μεταβλητές που έχουν τιμή *NULL* αυτές επιστρέφουν κενό, όπως ακριβώς και οι μεταβλητές boolean με τιμή *false*.

Χρήσιμες Συναρτήσεις Μεταβλητών Boolean

- *isset(όνομα μεταβλητής)*
 - ▣ Συνάρτηση που δείχνει αν μια μεταβλητή υπάρχει.
 - ▣ Επιστρέφει *true/false*.

```
<?php
```

```
$lyke = 1;
```

```
echo "Λύκε, Λύκε είσαι εδώ? " . isset($lyke);
```

```
?>
```



Λύκε, Λύκε είσαι εδώ? 1

Χρήσιμες Συναρτήσεις Μεταβλητών Boolean (2)

□ *unset(όνομα μεταβλητής)*

- ▣ Συνάρτηση που διαγράφει τα περιεχόμενα μιας μεταβλητής.
- ▣ Κάνει την τιμή της μεταβλητής ίση με *NULL*.

```
<?php
```

```
$me = "ΕΓΩ!!!!!!";
```

```
unset ($me);
```

```
echo "Ποιος είναι ο καλύτερος στην PHP??" . $me;
```

```
?>
```



Ποιος είναι ο καλύτερος στην PHP??

Χρήσιμες Συναρτήσεις Μεταβλητών Boolean (3)

- *empty(όνομα μεταβλητής)*
 - ▣ Ελέγχει αν μια μεταβλητή είναι άδεια.
 - ▣ Επιστρέφει *true/false*.

- Ως άδεια νοείται μια μεταβλητή
 - ▣ με τιμή *NULL* ή *false*.
 - ▣ με τιμή *0*.
 - ▣ με τιμή *""* ή *"0"*.
 - ▣ πίνακας χωρίς περιεχόμενα.
 - ▣ μεταβλητή ορισμένη χωρίς τιμή (*\$var1*).

Και μετά, και μετά??

1. Boolean  – Null (κενό)
2. Πίνακες

Πίνακες

- Ο πίνακας είναι τύπος δεδομένων που μπορεί να αποθηκεύσει περισσότερες από μια τιμές.
- Οι τιμές που παίρνει κάθε θέση του πίνακα μπορεί να είναι οτιδήποτε εμείς θέλουμε.
 - ▣ Αριθμός.
 - ▣ Αλφαριθμητικό.
 - ▣ Boolean.
 - ▣ ακόμα και άλλος πίνακας.

Κλειδιά - Αγκύλες

- Οι θέσεις ενός πίνακα ορίζονται με κλειδιά, τα οποία μπορεί να είναι
 1. Ακέραιοι αριθμοί όπως στους πίνακες του αντικειμενοστρεφούς προγραμματισμού.
 2. *Labels* (ετικέτες) που ορίζουμε εμείς.
- Με τις αγκύλες ορίζουμε την θέση του πίνακα που θέλουμε να πάρουμε ή να ορίσουμε την τιμή του
 - ▣ *\$array1[0] = 10; \$array2["Coyote"] = "HELP!!!"*
 - ▣ *echo \$array1[0] . \$array2 ["Coyote"]*

Ορισμός Πίνακα

- Ένας πίνακας ορίζεται με την εντολή `array()` και τις τιμές χωρισμένες με κόμμα (,)
 - ▣ Σε αυτή τη περίπτωση έχουμε δείκτες-κλειδιά ακέραιους (0, 1, 2 κτλ).

```
<?php
```

```
$array1 = array(2, "good to be", true);
```

```
echo $array1[0] . $array1[1] . $array1[2];
```

```
echo $array1;
```

```
?>
```



2 good to be 1
array

Ορισμός Πίνακα (2)

- Στην περίπτωση των *labels* έχουμε

```
<?php
```

```
$array1 = array("Babis" => "2 be",  
                "Soulara" => "or not 2 be");
```

```
echo $array1["Babis"] . $array1["Soulara"];
```

```
?>
```



2 be or not 2 be

Ορισμός Πολυδιάστατου Πίνακα

- Στην PHP αυτό επιτυγχάνεται βάζοντας πίνακα μέσα σε πίνακα

```
<?php
```

```
$array1 = array(array(9,10,11), array(4,5,6));
```

```
echo $array1[0][0] . " to " . $array1[1][1];
```

```
?>
```



9 to 5

Διαγραφή από Πίνακα

- Η διαγραφή ενός στοιχείου πίνακα γίνεται με την εντολή `unset`.
 - ▣ *`unset $array1[0];`*
- **ΠΡΟΣΟΧΗ!!!!** Μετά την διαγραφή του στοιχείου, τα υπόλοιπα στοιχεία μένουν στις θέσεις που είχαν, δεν μεταφέρονται δηλαδή προς τα αριστερά.

Παράδειγμα

```
<?php
```

```
$array1 = array("I", "am", "bad", "ass")
```

```
echo "$array1[0] $array1[1] $array1[2] $array1[3] \n";
```

```
unset $array1[2];
```

```
echo "$array1[0] $array1[1] $array1[2] $array1[3]";
```

```
?>
```



I am bad ass

I am ass



Συναρτήσεις Πινάκων

- `print_r()`: εμφανίζει τα περιεχόμενα ενός πίνακα.
 - ▣ Παίρνει σαν όρισμα την μεταβλητή – πίνακα.
 - ▣ Συνδυάζεται με πολύ καλά με την `<pre></pre>`.

```
<?php
```

```
$array1 = array(1,2,3);
```

```
print_r($array1)
```

```
?>
```



```
Array([0] => 1 , [1]=>2 , [2]=>3)
```

```
<pre>
```

```
<?php print_r($array1)
```

```
?>
```

```
</pre>
```



```
Array
```

```
(
```

```
[0] => 1 ,
```

```
[1] => 2 ,
```

```
[2] => 3
```

```
)
```

Συναρτήσεις Πινάκων (2)

- *count()*: επιστρέφει τον αριθμό των στοιχείων
- *max()*: επιστρέφει το μέγιστο στοιχείο
- *min()*: επιστρέφει το ελάχιστο στοιχείο
- *sort()*: ταξινομεί τα στοιχεία του πίνακα κατά αύξουσα σειρά
- *r_sort()*: ταξινομεί τα στοιχεία του πίνακα κατά φθίνουσα σειρά
- *in_array()*: βρίσκει αν μια τιμή υπάρχει στον πίνακα. Επιστρέφει *true/false*.

Συναρτήσεις Πινάκων (3)



- *implode()*: μετατρέπει έναν πίνακα σε string

```
$array1 = array("Damn", "you", "vile", "woman!");  
echo $string = implode(" ", $array1);
```



Damn you vile woman!

- *explode()*: μετατρέπει ένα string σε πίνακα

```
$string = "Victory is mine";  
$array1 = explode(" ", $string);  
<pre><?php print_r($array1)?>  
</pre>
```

Και πολλές άλλες!



```
Array(  
    [0] => "Victory" ,  
    [1] => "is" ,  
    [2] => "mine"  
)
```

Συνέχεια στην σελίδα 3...

1. Booleans – Null (κενό) ✓
2. Πίνακες ✓
3. Εντολή foreach

Εντολή *foreach*

- Η εντολή *foreach* χρησιμοποιείται πάντα με κάποιον πίνακα.
- Με χρήση αυτής, μπορούμε να προσπελάσουμε όλες τις θέσεις ενός πίνακα και να εκτελέσουμε τον κώδικα που περιλαμβάνεται στο σώμα της συνάρτησης για κάθε μια από τις τιμές του.

- Σύνταξη

```
foreach ( $array as $value ) {  
    // σώμα εντολών  
}
```

όπου η μεταβλητή *\$value* περιέχει τις τιμές του πίνακα.

Εντολή foreach (2)

- Εναλλακτική σύνταξη

```
foreach ( $array as $key => $value) {  
    // σώμα εντολών  
}
```

- *\$key* έχει τα κλειδί του τρέχοντος στοιχείου.
- *\$value* περιέχει τις τιμή του.

- Αν αλλάξει η τιμή του *\$value* μέσα στο σώμα, δεν αλλάζουν οι τιμές που έχει ο πίνακας.

Παράδειγμα

```
$heroes = array(  
    "Batman" => "AWESOME",  
    "Robin" => "gay"  
);  
foreach ( $heroes as $hero => $power ) {  
    echo " $hero is $power!! \n";  
}
```



*Batman is AWESOME!!
Robin is gay!!*



foreach Πολυδιάστατων Πινάκων

```
$matrix = array(
    array( 1, 2, 3 ),
    array( 4, 5, 6 ),
    array( 7, 8, 9 ),
);
foreach ( $matrix as $row ) {
    foreach ( $row as $element ) {
        echo "$element";
    }
    echo "\n";
}
```

Πολιτική

1. Booleans – Null (κενό) ✓
2. Πίνακες ✓
3. Εντολή foreach ✓
4. Διαχωρισμός σε Αρχεία

Διαχωρισμός σε Αρχεία

- Η PHP μας δίνει την πολύ χρήσιμη δυνατότητα να φορτώσουμε τον κώδικά ενός αρχείου σε ένα άλλο αρχείο.
- Με αυτόν τον τρόπο καταφέρνουμε να
 - ▣ Έχουμε καλύτερα οργανωμένο τον κώδικά μας
 - ▣ Επαναχρησιμοποιούμε κώδικα κατά βούληση
- Συνήθως χρησιμοποιείται για να
 - ▣ φορτώσουμε βιβλιοθήκες συναρτήσεων
 - ▣ φορτώσουμε header (κεφαλίδα) και footer (υποσέλιδο) στην σελίδα μας

Εντολή include

- Με την εντολή *include* φορτώνουμε εξωτερικά αρχεία στον κώδικά μας.
 - ▣ *include 'library.php'*
- Με αυτόν τον τρόπο τρέχουμε τον κώδικα του library.php
 - ▣ Είναι το ίδιο με το να κάνουμε copy/paste τον κώδικα που περιλαμβάνει το αρχείο που κάναμε *include*.
- Αν το αρχείο δεν βρεθεί τότε παράγει ένα **warning** και συνεχίζει.

Εντολή require

- Η εντολή require είναι παρόμοια σε χρησιμότητα με την include. Συντάσσεται ακρίβως το ίδιο.
 - ▣ *require 'library.php'*
- Η διαφορά των δύο εντολών έγκειται στο ότι με το require σε περίπτωση που δεν βρεθεί το αρχείο έχουμε **error** και όχι **warning**, καθώς και διακοπή της φόρτωσης της σελίδας.

include_once – require_once

- Σύνταξη
 - ▣ *include_once library.php;*
 - ▣ *require_once library.php;*
- Οι εντολές αυτές κάνουν το ίδιο πράγμα με τις αντίστοιχες που αναφέραμε προηγουμένως με μόνη διαφορά ότι τρέχουν τον κώδικα μόνο την πρώτη φορά που φορτώνει ένα αρχείο.
- Πολύ χρήσιμο σε περίπτωση που έχουμε πολλά αρχεία που μπορεί το ένα να κάνει *include* το άλλο.

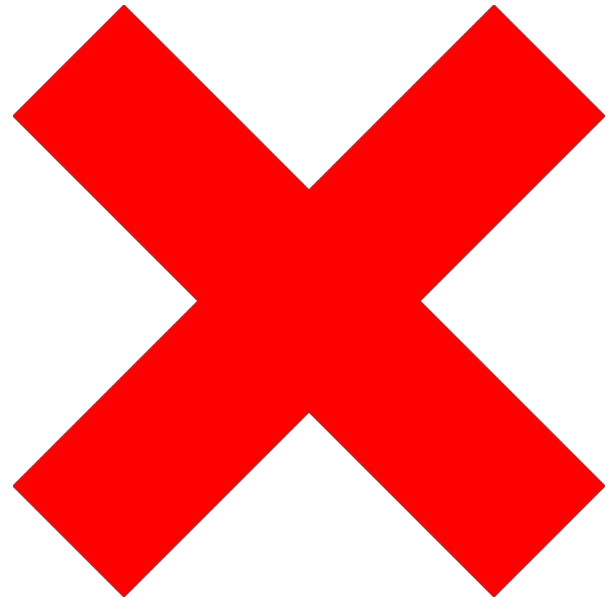
Αθλητικές Ειδήσεις

1. Booleans – Null (κενό) ✓
2. Πίνακες ✓
3. Εντολή foreach ✓
4. Διαχωρισμός σε Αρχεία ✓
5. Εμβέλεια Μεταβλητών

Εμβέλεια Μεταβλητών

- Εμβέλεια μεταβλητών ονομάζεται η περιοχή που ορίζονται (έχουν ισχύ) οι μεταβλητές.
- Παραδείγματος χάριν, οι μεταβλητές που βρίσκονται εκτός συναρτήσεων δεν ορίζονται σε αυτές

```
$var = 10;  
Function foo ( ){  
    echo $var;  
}
```



Λέξη-Κλειδί *global*

- Το προηγούμενο πρόβλημα μπορεί να επιλυθεί με την χρήση της λέξης-κλειδί *global*.
- Με την *global* δηλώνουμε ότι η μεταβλητή που χρησιμοποιούμε ορίζεται εκτός της συνάρτησης.

```
$var = 10;  
function foo ( )  
    global $var;  
    echo $var;  
}
```

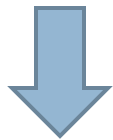


Εμβέλεια Μεταβλητών (2)

- Οι μεταβλητές που ορίζονται πριν από *include(_once)* και *require(_once)* μπορούν να χρησιμοποιηθούν στον κώδικα των αρχείων που φορτώνουμε.

```
$var = 10;
```

```
include 'library.php';
```



Το *\$var* είναι διαθέσιμο στο *library.php*

Μικρές Αγγελίες

1. Booleans – Null (κενό) ✓
2. Πίνακες ✓
3. Εντολή foreach ✓
4. Διαχωρισμός σε Αρχεία ✓
5. Εμφάνιση Μεταβλητών ✓
6. Συναρτήσεις Αρχείων

Συνάρτηση *opendir()*

- Συνάρτηση *opendir(όνομα φακέλου)*
 - ▣ Σαν όρισμα βάζουμε απλά το όνομα του φακέλου αν είναι μέσα στο ίδιο φάκελο με το αρχείο μας
 - ▣ Αλλιώς βάζουμε το path (σχετικό ή απόλυτο)
- Το αποτέλεσμα
 - ▣ Επιστρέφει ένα αντικείμενο τύπου φάκελος αρχείων
 - ▣ Επιστρέφει *false* αν δεν υπάρχει ο φάκελος
 - ▣ Χρησιμοποιείται από άλλες συναρτήσεις, που θα δούμε παρακάτω

Συνάρτηση *readdir()*

- Συνάρτηση *readdir()*
 - ▣ Δέχεται σαν όρισμα το αντικείμενο που επιστρέφει η *opendir()*.
- Το αποτέλεσμα
 - ▣ Επιστρέφει το επόμενο αρχείο μέσα στον φάκελο που επιλέξαμε.
 - ▣ Επιστρέφει *false* αν αποτύχει να επιστρέψει αρχείο.

Συνάρτηση *closedir()*

- Συνάρτηση *closedir()*
 - ▣ Δέχεται σαν όρισμα το αντικείμενο που επιστρέφει η *opendir()*.
- Το αποτέλεσμα
 - ▣ Κλείνει τον φάκελο που επιλέξαμε με την *opendir()* όταν τελειώσουμε τις εργασίες μας με αυτόν.
 - ▣ Επιστρέφει *false* αν αποτύχει να κλείσει τον φάκελο.

Συναρτήσεις Αρχείων

- Συνάρτηση *file_get_contents()*
 - ▣ Δέχεται σαν όρισμα ένα αντικείμενο τύπου αρχείο που επιστρέφει η *readdir()* ή ένα link σε αρχείο (απόλυτο ή σχετικό).
 - ▣ Επιστρέφει τα περιεχόμενα του αρχείου
 - ▣ Επιστρέφει *false* αν αποτύχει να διαβάσει το αρχείο.

- Συνάρτηση *file_put_contents()*
 - ▣ Δέχεται σαν όρισμα ένα αντικείμενο τύπου αρχείο που επιστρέφει η *readdir()* και τα περιεχόμενα που θέλουμε να του δώσουμε.
 - ▣ Αποθηκεύει τα περιεχόμενα στο αρχείο και σβήνει τα προηγούμενα.
 - ▣ Επιστρέφει *false* αν αποτύχει να διαβάσει το αρχείο.

Συναρτήσεις Αρχείων (2)

- Άλλες χρήσιμες συναρτήσεις

- ▣ *fopen()*

- ▣ *fclose()*

- ▣ *fwrite()*

- ▣ *fget()*

και άλλες πολλές.

<http://gr.php.net/manual/en/ref.filesystem.php>

Αναγγελίες Γάμων

1. Booleans – Null (κενό) ✓
2. Πίνακες ✓
3. Εντολή foreach ✓
4. Διαχωρισμός σε Αρχεία ✓
5. Εμφάνεια Μεταβλητών ✓
6. Συνάρτησεις Αρχείων ✓
7. Διαχείριση Αρχείων

Ανέβασμα Αρχείων

- Τα αρχεία στέλνονται στον server με αίτημα *POST*.
- Αποθηκεύονται προσωρινά στην μνήμη όσο τρέχει το πρόγραμμά μας.
- Πρέπει να τα αποθηκεύσουμε αλλιώς χάνονται με το πέρας της εκτέλεσης.

```
<form enctype="multipart/form-data" method="post">
```

Send this file:

```
<input name="userfile" type="file" />
```

```
<input type="submit" value="Send File" />
```

```
</form>
```

Μεταβλητή `$_FILES`

- Η μεταβλητή `$_FILES` περιέχει όλα τα αρχεία που ανέβασε ο χρήστης.
- `$_FILES['userfile']['name']`
 - ▣ Περιέχει το πραγματικό όνομα κάθε αρχείου.
- `$_FILES['userfile']['tmp_name']`
 - ▣ Περιέχει το προσωρινό όνομα κάθε αρχείου.
- `$_FILES['userfile']['size']`
 - ▣ Περιέχει το μέγεθος κάθε αρχείου σε bytes.

Αποθήκευση Αρχείων

- Συνάρτηση αποθήκευσης uploaded αρχείων
 - ▣ *move_uploaded_files(\$filename, \$destination)*
 - ▣ γίνεται έλεγχος πριν την αποθήκευση ότι το αρχείο προέρχεται από ανέβασμα.
 - ▣ η μεταβλητή *\$filename* είναι το αρχείο που ανέβηκε.
 - ▣ η μεταβλητή *\$destination* είναι εκεί που θα αποθηκευτεί.

```
<?php
```

```
    $destination = 'C:\Uploads\'
```

```
    if( !empty( $_FILES ) ) {
```

```
        $destination .= $_FILES[ 'foo' ][ 'name' ];
```

```
        $filename = $_FILES[ 'foo' ][ 'tmp_name' ];
```

```
        move_uploaded_file( $filename, $destination );
```

```
    }
```

```
?>
```

Επικήδριοι

1. Booleans – Null (κενό) ✓
2. Πίνακες ✓
3. Εντολή foreach ✓
4. Διαχωρισμός σε Αρχεία ✓
5. Εμφάνεια Μεταβλητών ✓
6. Συνάρτησεις Αρχείων ✓
7. Διαχείριση Αρχείων ✓
8. Headers (Κεφαλίδες)

Συνάρτηση *header()*

- Η συνάρτηση *header()* παίρνει ως όρισμα ένα String και το στέλνει στην κεφαλίδα της απάντησης.
- Πρέπει να τρέξει πριν από οποιοδήποτε output
 - ▣ Είτε είναι html.
 - ▣ Είτε space, tab ή enter.

<html>

<?php

header('Location: http://www.apisti.com/');

?>

Χρήσιμα Headers

- *header('Location: http://www.apisti.com/')*
 - Κάνει redirect σε μια διεύθυνση.
 - Η διεύθυνση πρέπει να είναι απόλυτη.
- *header('Content-type: text/html; charset=utf-8')*
 - Είναι εντολή ισοδύναμη με τον κώδικα

<meta

http-equiv="Content-type"

content="text/html; charset=utf-8"

/>

Ζώδια

1. Booleans – Null (κενό) ✓
2. Πίνakes ✓
3. Εντολή foreach ✓
4. Διαχωρισμός σε Αρχεία ✓
5. Εμφάνιση Μεταβλητών ✓
6. Συνάρτησεις Αρχείων ✓
7. Διαχείριση Αρχείων ✓
8. Headers (Κεφαλίδες) ✓
9. Cookies (Μπισκότα) – Session (Συνοδοί)



Cookies (Μπισκότα)

- Χρησιμοποιούνται για να μεταφέρουν προσωπικά στοιχεία του χρήστη από σελίδα σε σελίδα ή να τα αποθηκεύουν ώστε να μην χρειάζεται να τα εισάγει ο χρήστης συνέχεια.
 - ▣ Στην πραγματικότητα, είμαι μια λίστα της μορφής κλειδί → τιμή.
 - ▣ Έχουν ημερομηνία λήξης.
 - ▣ Αποθηκεύονται στον υπολογιστή του χρήστη.
 - ▣ Αποστέλλονται με κάθε αίτημα.
 - ▣ Για κάθε domain έχουμε διαφορετικό *cookie*.
- Χρησιμοποιούνται κατά κύριο λόγο για να διαχωρίζουν τους χρήστες μεταξύ τους.
 - ▣ Όσο περισσότερα έχουμε, τόσο **μεγαλύτερο overhead**.
 - ▣ Αυτό τα κάνει πολύτιμα. Αν κάποιος κλέψει το δικό σου, γίνεται «εσύ».

Γραφικό Παράδειγμα



Yo, bitches!!!



Wazzup? You are Monty for now on.

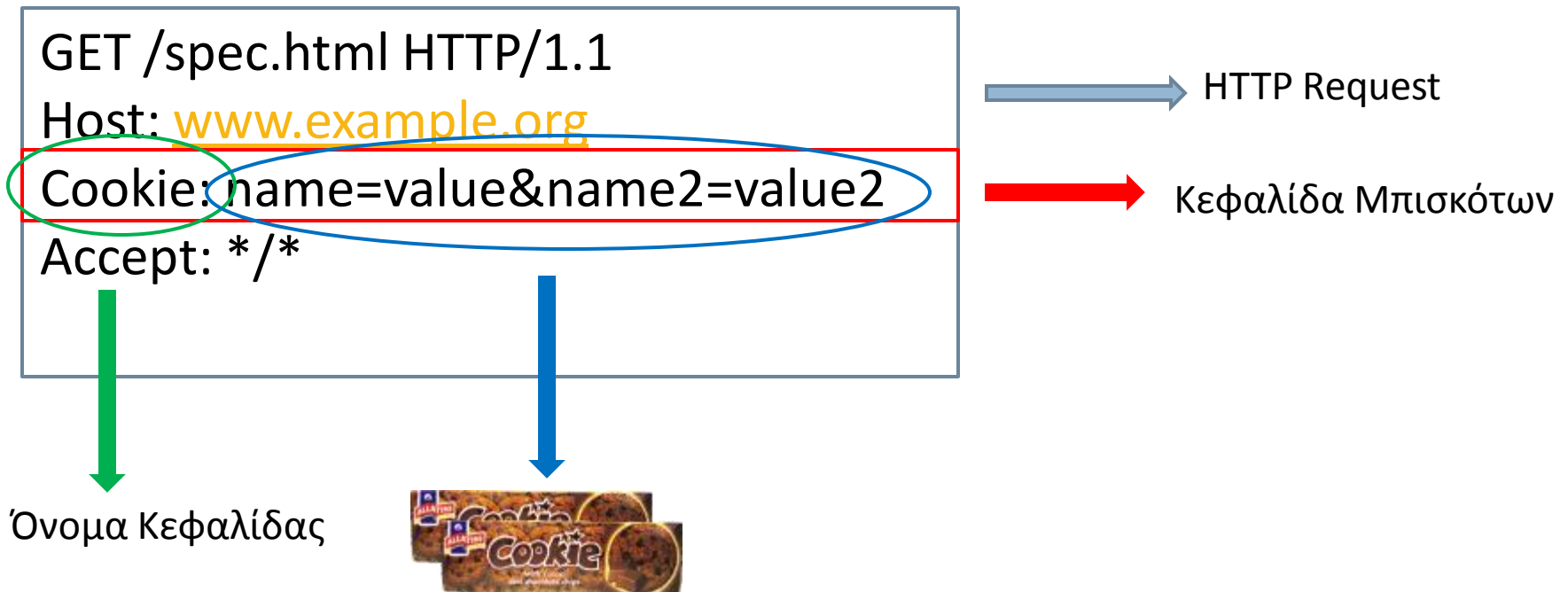


Monty is in the house!!!



Recognise!!!

Πραγματικό Παράδειγμα



Εδώ έχουμε δύο μπισκότα

1. Το μπισκότο *name* με τιμή *value*
2. Το μπισκότο *name2* με τιμή *value2*

Cookies (2)

- Τα *cookies* μεταφέρονται στις κεφαλίδες των αιτημάτων.
 - ▣ Αυτό σημαίνει ότι πρέπει να τοποθετηθούν πριν από όλα τα output (τα είπαμε πριν λίγο αυτά).
- Σύνταξη *setcookie(\$name, \$value, \$expire)*
 - ▣ *\$name*: το όνομα της μεταβλητής του cookie.
 - ▣ *\$value*: η τιμή της μεταβλητής του cookie.
 - ▣ *\$expire*: δείχνει πότε λήγει το cookie.

Παραδείγματα

- `setcookie("name" , "alitiz" , 0)`
 - ▣ Η μεταβλητή `name` παίρνει την τιμή `"alitiz"` και το `cookie` θα λήξει όταν κλείσουμε τον browser.

- `setcookie("name" , "alitiz" , time() + 3600)`
 - ▣ Η συνάρτηση `time()` επιστρέφει την ώρα του υπολογιστή.
 - ▣ Σε αυτή τη περίπτωση θα κρατήσει το `cookie` για μια ώρα ακόμα.

- `setcookie("name" , "alitiz" , time() - 3600)`
 - ▣ Με αυτόν τον τρόπο σβήνουμε τα `cookies` που δεν χρειαζόμαστε πλέον.

Μεταβλητή \$_COOKIE

- Σε αυτή τη μεταβλητή είναι αποθηκευμένα όλα τα *cookies* μας.
 - ▣ Για κλειδιά έχει τα ονόματα των *cookies*.
 - ▣ Για τιμές έχει τις τιμές των *cookies*.

<?php

```
$expire = time() + 60 * 60 * 24 * 7;  
setcookie( "visited", "true", $expire );
```

```
if ( isset( $_COOKIE[ "visited" ] ) ) {  
    echo "Yo Bro!";
```

```
}  
else {  
    echo "Get the fuck out of my house!!!";  
}
```

?>

Sessions (Συνοδοί)

- Η PHP για να μας κάνει την ζωή πιο εύκολη χρησιμοποιεί τα *sessions*.
 - ▣ Αντί να έχουμε πολλά *cookies* έχουμε ένα.
- Το *cookie* αυτό λέγεται *SessionID*
 - ▣ Είναι ένας τυχαίος ακέραιος αριθμός
 - ▣ Διαφορετικός για κάθε χρήστη.
 - ▣ Ορίζεται την πρώτη φορά που ανοίγει κάποιος μια σελίδα.
- Με αυτόν τον τρόπο αποθηκεύονται στον server οι μεταβλητές.
 - ▣ Έπειτα, η διαχείρησή τους γίνεται από την PHP.

Sessions (2)

- Με την συνάρτηση *session_start()*
 - ▣ Εκκινούμε ένα *session* για κάποιον χρήστη.
 - ▣ Επαναφέρουμε το *session* που αντιστοιχεί στο χρήστη σε περίπτωση που στείλει έγκυρο ID.
- Μετά μπορούμε να χρησιμοποιήσουμε την μεταβλητή *\$_SESSION*
 - ▣ Περιλαμβάνει τις μεταβλητές που αντιστοιχούν στο *SessionID* του χρήστη.

Γραφικό Παράδειγμα



Yo, bitches!!!



Wazzup? You are 444 for now on.



217 - Naruto
666 - Devil

SID - 444



444 is in the house!!! My name is 50Cent!



Recognise!!!



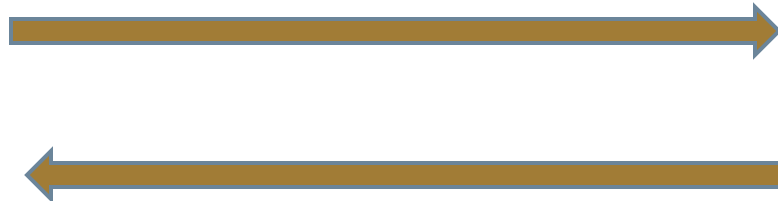
217 - Naruto
666 - Devil
444 - 50Cent

Γραφικό Παράδειγμα (2)

SID - 444



Bro, it's 444!



Yo 50! Respect!

217 - Naruto
444 - 50Cent
666 - Devil



SID - 999



Hi, I am 999!



Who the fuck are you, white-ass bitch???

217 - Naruto
444 - 50Cent
666 - Devil



Παράδειγμα

```
<?php
    session_start();
    if ( isset( $_SESSION[ 'pageviews' ] ) ) {
        ?><p>Έχεις δει τη σελίδα <?php
        echo $_SESSION[ 'pageviews' ];
    ?> φορές</p><?php
        ++$_SESSION[ 'pageviews' ];
    }
    else {
        $_SESSION[ 'pageviews' ] = 1;
    }
?>
```

ΟΥΦ! ΕΠΙΤΕΛΟΥΣ ΤΕΛΕΙΩΣΕ!!!

- Τώρα μπορείτε να κάνετε πολλά πράγματα σε ΡΗΡ!!! (Άντε να δούμε)...

THE END!!!

- Ευχαριστούμε πάρα πολύ για την προσοχή σας

